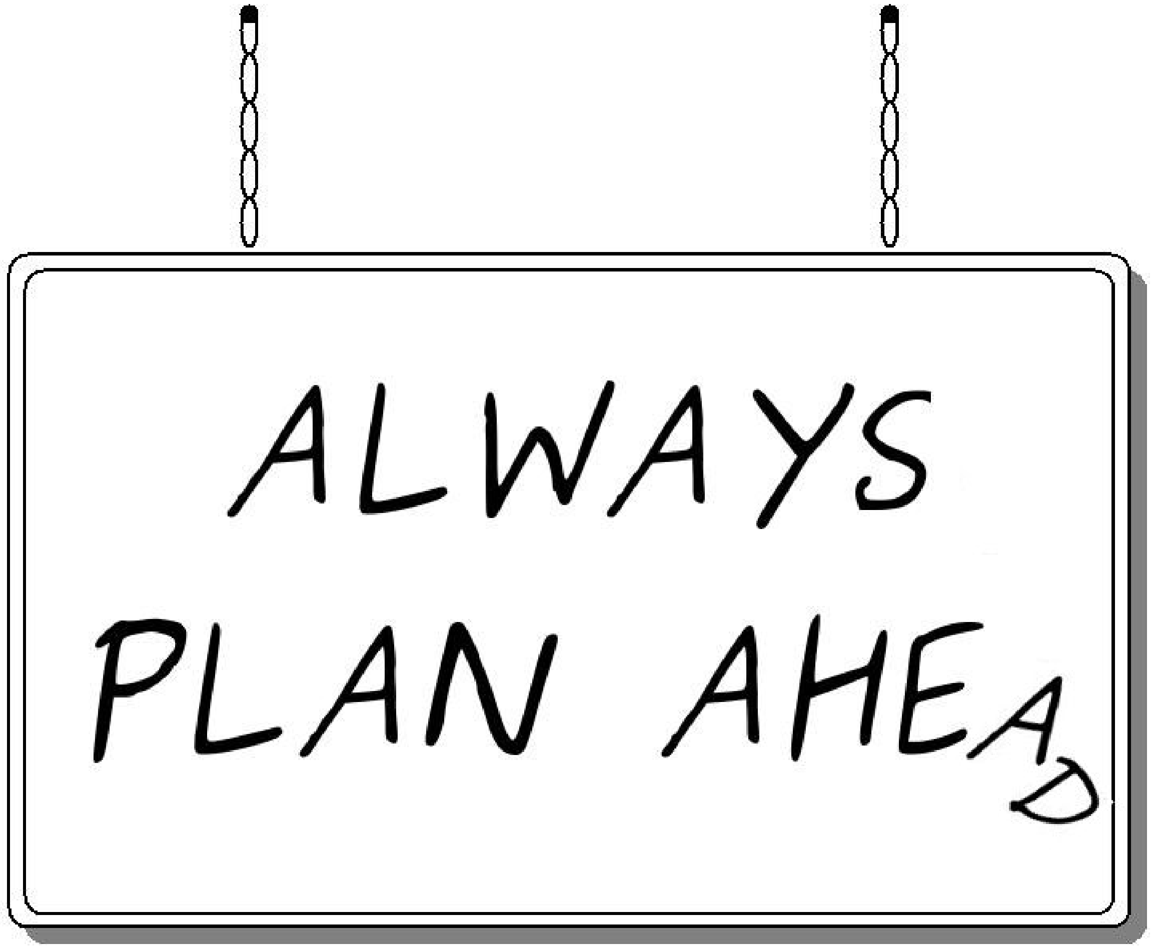
SOFTWARE ENGINEERING PROCESSES AND MANAGEMENT



WORKSHOP 4
MEET THE PROJECT PLAN

INTRODUCTION

The aim of this workshop is to gain experience at using formal planning and scheduling processes, and to explore the relationship between software development lifecycle processes, team structures, planning, and scheduling.

DEVELOPING A SCHEDULE

When developing a schedule, the following process model is generally adopted:

1. **Identify the tasks** — The first step is to identify the tasks. Starting from the phases in the development lifecycle, we compartmentalise the project using a work breakdown, or similar tool, until the leaf nodes are *manageable* tasks.

2. **Identify the dependencies** — The next step is to identify the dependencies between the tasks, and model these dependency using a task network, or similar tool.

3. **Estimate resources and durations of tasks** — The third step is to estimate the resources and durations of each of the tasks. We will cover this in more detail later in the course, but they important information includes the duration of the tasks, and any resources that the task uses or produces, such as staff, hardware, or project artifacts.

4. **Construct the schedule** — Using the task network and estimates of duration & tasks, construct a schedule for the project. The schedule must identify the start and end dates for every task, relative to the start date of the project, which resources (including people) will be employed on those tasks.

5. **Wait(1 week); Goto 1** — Planning and scheduling does not stop there. The schedule must be revisited throughout the project — for example, every week — to take into account new information. On large projects, it is not uncommon to have a team of full-time schedulers.

The first three steps in the process are highly important. It is not uncommon for inexperienced managers to sit down and develop what looks like a schedule without doing task & dependency analysis, or estimates of resources & duration. However, the resulting schedule will be more a work of fiction than a useful tool for planning.

TIPS FOR SCHEDULING

The following are some tips that may come in useful when working out a schedule for a project.

1. **Schedule small tasks** — If tasks in the schedule are too large, it is difficult to see how they will be achieved. Aim for tasks that are measured in days or weeks, rather than months. If a task is scheduled to last for months, break the task down into smaller milestones that are easier to manage and understand.

2. **Prioritise tasks that add value to the customer** — A recent study by the Standish Group[1] found that around 45% of functions in most applications were never used. By prioritising tasks based on the value given to the user/customer, a valuable product will emerge earlier.

[1] *What Are Your Requirements?*, Standish Group International, Inc., 2003, Standish Group.

3. **Assume perfect resources initially** — When defining the schedule initially, one way to deal with the complexity is to assume that you have sufficient resources, including staff, to run any non-dependent tasks simultaneously. Once you have a schedule that works under this assumption, start removing resources and re-working the schedule to fit the new constraints. A good idea is to first find resources that can be removed/replaced without affecting the schedule.

4. **Plan as a team** — When constructing a plan, involve the people that will be performing the tasks. Collectively, the team will have a better understanding of the dependencies than any one person, and furthermore, they are motivated to get the schedule right, because they will be directly affected by anything that goes wrong. This does not mean one the entire team should sit and write the plan together, however, all team members should have input, and should accept the schedule if possible.

5. **Include risk management in your planning** — Risk management, which will be discussed later in the subject, should be considered at every stage of a project, especially planning. While we have only scratched the surface of risk management so far, we have been discussing the risks associated with various software engineering activities.

PLANNING AND SCHEDULING FOR A PROJECT

Table 1 contains a work breakdown for the initial requirements and planning phases of the language technology project, including the resource usage (people only) and duration of each task. Using the information in this table, your tasks for this week's workshop are:

Task network (20 mins). Develop a task network by identifying the dependencies between the tasks. A column has been left empty in the table for you to annotate should you wish to use this. There are some dependencies that you will need to work out yourself, however, we will give you the following information to start.

- Tasks 1.2 (initial research) and 1.3 (problem definition) are independent.
- Task 2.2.1 (second pass of requirement elicitation) depends on task 2.1.2 (first pass requirements analysis).
- Task 3.1 (technical risk assessment) depends on task 2.1.2 (first pass requirements analysis) but not on any task in 2.2.
- Task 2.4 (requirements validation) depends on task 2.3 (requirements specification).
- Task 2.5 (requirements sign off) depends on task 3.1 (technological risk assessment) and 2.4 (requirements validation).

Schedule (15 mins). Using the resource and duration estimates (the first options), and your task network, construct a PERT chart for this phase of the project. Identify the critical path.

Planning (15 mins). Assume 8 people working on the project, each of whom can perform any task. Using both the first and second options for the estimates in Table 1, can you schedule the tasks such that the project can be completed in 28 days?

					Option 1		Option 2	
Work Breakdown				Dependencies	People	Days	People	Days
1.	Concept Phase							
	1.1	Concept Planning			2	1		
	1.2	Initial Research			4	4	6	3
	1.3	Problem Definition with Client			2	1		
	1.4	Initial Project Plan			2	1		
2.	Requirements							
	2.1	Requirements Iteration 1						
		2.1.1	Requirements Elicitation first pass		3	2		
		2.1.2	Requirements Analysis first pass		4	3	6	2
		2.1.3	Requirements Model		4	3	6	2
	2.2	Requirements Iteration 2						
		2.2.1	Requirements Elicitation second pass		4	3		
		2.2.2	Requirements Analysis second pass		4	3	6	2
		2.2.3	Requirements Model		4	4	6	3
	2.3	Requirements Specification			4	5		
	2.4	Requirements Validation			4	4		
	2.5	Requirements Sign-off			2	4	4	3
3.	Project Planning							
	3.1	Technological Risk Assessment			2	4	4	3

Table 1: The initial requirements and planning phases of the language technology project.